REMARKS

Claims 1, 4-10, 12-16 and 19-22 were pending at the time of examination. Claims 1, 8-10, 16 and 21 have been amended. No new matter has been added. The applicants respectfully request reconsideration based on the foregoing amendments and these remarks.

Claim Rejections – 35 U.S.C. § 103

Claims 1, 4-6, 10, 12-14, 16, 19, 21 and 22 were rejected under 35 U.S.C § 103(a) as being unpatentable over U.S. Patent No. 5,991,173 to Unger et al. (hereinafter "Unger") in view of Ainon "Storing text using integer codes," 1986, *Proceedings of the 11ᵗʰ conference on Computational linguistics* (hereinafter Ainon); and further in view of Storer et al., "The Macro Model for Data Compression," 1978, *Proceedings of the 10ᵗʰ Annual ACM symposium on Theory of computing* (hereinafter Storer). The applicants respectfully traverse this rejection.

Claim 1 has been amended to specifically state that the method refers to compiling a computer source program, which is written in a high-level programming language and to further define how the differential name is generated, in order to more clearly distinguish the applicants' invention from the techniques described in Unger, Ainon and Storer. More specifically, the differential name for the encoded program symbol name is generated relative to a base symbol that identifies a containing scope for the program symbol. The containing scope can be, for example, a namespace, a package, a module, a container object, or a function, which all can define an "environment" in which the encoded programming symbol operates. Unger, Ainon and Storer, on the other hand, describe various methods of text compression. Neither Unger, nor Ainon nor Storer, alone or in combination, reasonably teach or suggest the subject matter of claim 1, as amended.

More specifically, claim 1, as amended, recites the step of:

> "generating a differential name for the encoded program symbol name relative to a base symbol identifying a containing scope for the program symbol, wherein the containing scope is selected from a group consisting of: a namespace, a package, a module, a container object, and a function, the differential name having a reduced-size format as compared to the encoded program symbol name;"

Neither Unger, nor Ainon nor Storer, alone or in combination, reasonably teach or suggest this step for at least the following reasons. Unger's method of compression compares the words of a text file (including *numeric strings, decimal points, currency symbols,* etc.) with the one or more predetermined dictionaries, and then with a supplemental dictionary if the words

Attorney Docket No.: SUN1P380/P4501   Page 7 of 10   Serial No.: 09/649,270

PAGE 10/13 * RCVD AT 1/13/2005 6:48:23 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/2 * DNIS:8729306 * CSID:16509618301 * DURATION (mm-ss):04-06

cannot be found in the first dictionary (*see* col. 10, lines 23-39). Numeric characters can be encoded with a special predetermined "numeric" dictionary (*see id.*). As such, the symbols described in Unger are merely characters in a text file and have no function or meaning outside of the text file being compressed. Furthermore, the symbols in Unger have no "containing scope," since they are all characters in a text file. In contrast, the program symbol names of the applicants' invention are not simply characters in a text file, but are more correctly characterized as programming references or pointers with meaning outside of the context of a text file. Furthermore, in Unger (as well as in Ainon and Storer), the goal is to encode as much as possible of a text file to achieve a maximum compression ratio, whereas in the applicants' invention, only the program symbol names are encoded and the non-program symbol information is left unaltered.

Ainon does not generate a <u>differential name relative to a containing scope</u>, as required by the claim limitation, but instead a two-byte integer code that represents a position in a word list where the word in question is stored. The fact that the word list is arranged by base words and various forms thereof merely facilitates finding the proper two-byte integer code for a given word in a text to be coded, but does not suggest the notion of containment or generating a differential name relative to a containing scope. In addition, claim 1 requires that the differential name have a <u>reduced-size format</u> as compared to the encoded program symbol name. Many words can be represented with two bytes or less. Thus, the two-byte numerical representation of a word is also not necessarily of reduced-size format compared to a word that is encoded, as is required. Also, in Ainon, there is always a one to one correspondence between a word and its encoded two-byte integer code representation, whereas in the applicants' invention the differential name may vary based on the containing scope for the encoded program symbol name. Furthermore, in the applicants' invention, the differential names are generated for <u>encoded program symbol names</u>, which is not even remotely similar to coding words in a text.

With respect to Storer, the applicants respectfully contend that replacing parts of strings with pointers is no more similar to the claimed limitation than replacing words with two-byte integer codes, as in Ainon. Rather, Storer merely describes a method for reducing the length of strings. The strings do not describe program symbols and do not have any context dependency. Thus, Unger, Ainon and Storer merely disclose alternative ways of encoding text, neither of which, alone or in combination, anticipates or render claim 1 obvious.

For at least the above reasons, the rejection of claim 1 is unsupported by the cited art and should be withdrawn.

Claim 16 is a *Beauregard* claim corresponding to claim 1. For reasons substantially similar to those set forth above, the applicants respectfully contend that the rejection of claim 16 is unsupported by the cited art and should be withdrawn.

Claims 4-9 depend from claim 1, and the rejections of these claims are unsupported by the cited art for at least the reasons discussed above with regards to claim 1, and should be withdrawn.

Claims 19-20 depend from claim 16, and the rejections of these claims are unsupported by the cited art for at least the reasons discussed above with regards to claim 16, and should be withdrawn.

Claim 10 describes a method for generating encoded program symbol names in an uncompressed form, and was rejected for the same rationale that was set forth in the rejection of claim 1. Claim 10 contains limitations relating to program symbol names, base symbols, and differential program symbol names and formats. Consequently, for at least the reasons discussed above with regards to claim 1, the applicants respectfully contend that the rejection of claim 10 is unsupported by the cited art and should be withdrawn.

Claim 12 depends from claim 10, and the rejection of this claim is therefore unsupported by the cited art for at least the same reasons, and should be withdrawn.

Claim 21 is a *Beauregard* claim corresponding to claim 1. For reasons substantially similar to those set forth above, the applicants respectfully contend that the rejection of claim 16 is unsupported by the cited art and should be withdrawn.

Claim 13 was rejected for substantially the same reasons as claim 1. Claim 13 includes the limitation that the enhanced compiler includes "one or more differential names corresponding to the program symbol names." The program symbol names and the differential names have been discussed above with respect to the rejection of claim 1. For reasons substantially similar to those set forth above with regards to claim 1, the applicants respectfully contend that the rejection of claim 13 is unsupported by the cited art and should be withdrawn.

Claims 14 and 15 depend from claim 13, and the rejections of these claims are unsupported by the cited art for at least the reasons discussed with regards to claim 13, and should be withdrawn.

Conclusion

The applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP
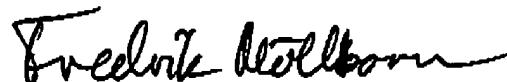
Fredrik Mollborn
Reg. No. 48,587

P.O. Box 778
Berkeley, CA 94704-0778
(650) 961-8300